

Exercițiul 5

Îmbunătățiri:

1. Adăugarea unui contor de acces pentru fiecare fir de execuție:

```
private static AtomicIntegerArray accessCount = new  
AtomicIntegerArray(n);
```

2. Verificarea contorului înainte de avansarea la un nou nivel:

```
if (k != i && level.get(k) >= L && victim.get(L) == i &&  
accessCount.get(i) > accessCount.get(k))
```

În plus față de condițiile inițiale, un thread așteaptă și dacă a avut acces mai des decât un alt thread la același nivel (sau cu un nivel mai înalt). Aceasta asigură că niciun thread nu accesează secțiunea critică mai frecvent decât altele.

Fairness integrat – În acest mod, prin rotația bazată pe *accessCount*, implementăm fairness fără o metodă externă de verificare. Algoritmul permite accesul echitabil, deoarece fiecare thread trebuie să aștepte până când toate celelalte au accesat secțiunea critică de un număr aproximativ egal de ori.

Astfel, această modificare asigură 0-bounded waiting și fair access la secțiunea critică.